How to Compare Profiled Side-Channel Attacks?

François-Xavier Standaert[1]*, François Koeune[1]**, Werner Schindler[2]

[1] UCL Crypto Group, Université catholique de Louvain, B-1348 Louvain-la-Neuve.
[2] Bundesamt für Sicherheit in der Informationstecknik (BSI), 53175 Bonn, Germany.

fstandae,francois.koeune@uclouvain.be; werner.schindler@bsi.bund.de

Abstract. Side-channel attacks are an important class of attacks against cryptographic devices and profiled side-channel attacks are the most powerful type of side-channel attacks. In this scenario, an adversary first uses a device under his control in order to build a good leakage model. Then, he takes advantage of this leakage model to exploit the actual leakages of a similar target device and perform a key recovery. Since such attacks are divided in two phases (namely profiling and online attack), the question of how to best evaluate those two phases arises. In this paper, we take advantage of a recently introduced framework for the analysis of side-channel attacks to tackle this issue. We show that the quality of a profiling phase is nicely captured by an information theoretic metric. By contrast, the effectiveness of the online key recovery phase is better measured with a security metric. As an illustration, we use this methodology to compare the two main techniques for profiled side-channel attacks, namely template attacks and stochastic models. Our results confirm the higher profiling efficiency of stochastic models when reasonable assumptions can be made about the leakages of a device.

1 Introduction

Side-channel attacks are a powerful class of cryptanalysis techniques in which an adversary not only takes advantage of the mathematical properties of an algorithm but also of the physical properties of its implementation. Profiled side-channel attacks are the most powerful type of side-channel attacks and can be viewed as divided in two phases. First, a profiling phase provides an adversary with a training device and allows him characterizing its physical leakages. Second, an online exploitation phase is mounted against a similar target device in order to perform a key recovery. Standard profiled side-channel attacks include template attacks and stochastic models, respectively introduced in [1] and [5].

Because of their division in two phases, a usual question for such attacks is to determine their effectiveness in profiling and attacking a device. In this work, we follow the analysis of [2] in which the performances of template attacks and stochastic models were analyzed. In this reference, the efficiency of the online phase was nicely captured by measuring the success rate of a key recovery

* Associate researcher of the Belgian Fund for Scientific Research (FNRS - F.R.S.)

** Supported in part by the Walloon Region research project SCEPTIC.

adversary exploiting templates or stochastic models. By contrast, the criteria used to quantify the quality of the profiling phase had a more ad hoc flavor. As a consequence, we suggest that the framework of [7] can be used to improve this analysis. We present experiments to confirm how and why an information theoretic metric captures the profiling efficiency of an attack while a security metric rather measures the effectiveness of its online phase. Hence, our results confirm the previous intuitions with a more rigorous theoretical background. In practice, we observe that stochastic models built from sound engineering assumptions can give a very precise image of a device's leakages from a reduced amount of profiling measurements. More formally, our experiments can be viewed as the practical counterparts of Theorems 1 and 2 in [7]. They show that the proposed principles for comparing side-channel attacks are not only theoretical but can also be practically meaningful and solve actual engineering problems.

The rest of this paper is structured as follows. Section 2 introduces the preliminary assumptions in profiled side-channel attacks. Section 3 recalls the evaluation metrics of [7]. Section 4 provides a brief description of the template attacks and stochastic models with a discussion of their parameters. The core of the paper is in Sections 5 and 6 in which our experimental comparisons are presented and their limitations are analyzed. Eventually, conclusions are in Section 7.

2 Preliminary assumptions for profiled attacks

Before starting a careful analysis of particular types of attacks, it is important to consider the different assumptions that can sometimes be hidden in the description and implementation of a profiled side-channel attack. In particular, this section aims to list four decisions that generally have to be taken.

Known or chosen plaintext models. As a matter of fact, any profiled side-channel attack starts by building a leakage model that will be used in the online part of an attack to predict the actual leakages of a target device. As a consequence arises the question: "for which inputs will the model be built?". In the practice of side-channel attacks, there are essentially two available choices, namely known or chosen plaintext leakage models. If a *chosen plaintext leakage model* is decided, it suffices if the adversary only builds a model for certain plaintexts or sequences of plaintexts. Hence, the same chosen plaintexts or sequences of plaintexts will have to be used in the online phase of the attack. By contrast, if a *known plaintext leakage model* is considered, the leakages corresponding to the encryption of any plaintext can be exploited in the online phase of the attack[1].

[1] We mention that chosen plaintext *models* are not particularly desirable in template attacks (since they limit the exploitable plaintexts in the online phase of the attack). But they should not be confused with (possibly adaptive) chosen plaintext *attacks* that generally improve the effectiveness of the online phases. Note also that known or chosen ciphertexts could be considered equivalently.

Weight or distance based models. A side-channel adversary always has to do some minimal assumptions on the architecture of his target device. Typically, side-channel leakages such as the power consumption are generally dependent on the transition between two inputs rather than on single inputs. If such transition-based leakages are actually observed, it implies that the models also have to be built for different input transitions rather than for different inputs. Such a context typically corresponds to the Hamming distance leakage models described, *e.g.* in [4]. By contrast, in certain devices (*e.g.* smart cards) the meaningful transitions are not between two inputs but between a variable input and a constant state. In such scenarios, leakage models based on single inputs are again meaningful, just as when Hamming weight models apply[2].

Symmetry properties in the leakages. Depending on the two previous decisions, an adversary will decide to build a model (*e.g.* templates) for different inputs of the target device. In the context of a block cipher, it means that models have to depend on plaintexts and keys. But the lower the number of templates to build, the better the profiling efficiency. Hence, one will typically try to take advantage of symmetry properties in the leakages, such as the Equal Images under different Subkeys (EIS) property defined in [5]. For example, if it is known that (most of) the leakages of a block cipher implementation are not dependent on both the plaintext and the key but only on the XOR between the plaintext and the key, then templates can be built only for these XOR values. For further considerations on symmetries, we refer the interested reader to [6].

Need to program a target device. Eventually, it is worth mentioning that it is generally assumed that profiled side-channel attacks require a device that one can program (*e.g.* control the keys) during profiling. In fact, if an EIS property is assumed, it can be sufficient to profile the device with only one known key. When stochastic models are considered, it may even be possible to profile without a device for which the key is known (see [5], Remark 2 for the details).

2.1 Target implementation

The goal of this paper is not to investigate one particular device but to provide a methodological contribution to the comparison of profiled side-channel attacks. For this reason, we decided to analyze a simple simulated attack scenario in which all the parameters are under control. As will be clear later, it allows putting forward interesting intuitions on the respective effectiveness of the template attacks and stochastic models but also on their limitations.

In practice, we investigated the following context. Let k be the first master key byte of the AES Rijndael and x_i be a corresponding input plaintext byte. Let

[2] In theory, longer history effects could be observed, *i.e.* the actual leakages may not only depend on the transition between two inputs but also on previous ones. We focus on the weight and distance based models because they are very common in the literature. But extending the choice towards other cases would be possible.

S be the AES S-box and $y_i = \mathsf{S}(x_i \oplus k)$ be the output of this S-box. We consider an adversary that is provided with leakage traces[3] of the form $[x_i, H_W(\mathsf{S}(x_i \oplus k)) + n_i]$ where H_W is the Hamming weight function and n_i is a realization of normally distributed noise, described by a random variable N_i with expectation $\mu = 0$ and with variance σ^2. In the following sections, we will evaluate this adversary in function of two parameters: the amount of traces used in the profiling stage of the attack q_p and the amount of traces used in the online phase of the attack q. With respect to the previous assumptions, we will build known plaintext models assuming weight based leakages. Eventually, the adversary will take advantage of an EIS property and assume that the leakage for every pair (x_1, k_1), (x_2, k_2) such that $x_1 \oplus k_1 = x_2 \oplus k_2$ is identical. We acknowledge that this scenario (mainly selected for tutorial purposes) hides the practical problem of selecting the meaningful time samples in the leakage traces (discussed, *e.g.* in [2, 9]), due to its univariate nature. However, the proposed evaluation methodology can be straightforwardly extended to multivariate probability distributions.

3 Evaluation metrics

Following the framework introduced in [7], we will evaluate our different experiments with a combination of information theoretic and security metrics.

Information theoretic metric. Let K be a discrete random variable representing the target key byte of our side-channel attacks and k be a realization of this variable (*i.e.* the key in one instance of attack). Let $\mathbf{L}_q$ be a random vector describing random side-channel observations generated with q queries to the target physical computer and $\mathbf{l}_q = [l_1, l_2, \ldots, l_q]$ be a realization of this random vector, with *e.g.* $l_i = H_W(\mathsf{S}(x_i \oplus k)) + n_i$ (and $L_i = H_W(\mathsf{S}(x_i \oplus k)) + N_i$) as explained in the previous section. Let finally $\Pr[k|\mathbf{l}_q]$ be the conditional probability of a key byte k given a leakage $\mathbf{l}_q$. We define a conditional entropy matrix as:

$$\mathbf{H}^q_{k,k^*} = -\sum_{\mathbf{l}_q} \Pr[\mathbf{l}_q|k] \cdot \log_2 \Pr[k^*|\mathbf{l}_q], \tag{1}$$

where k^* denotes a possible key class candidate in the attack. From this matrix, we derive Shannon's conditional entropy as follows:

$$\mathrm{H}[K|\mathbf{L}_q] = -\sum_{k} \Pr[k] \sum_{\mathbf{l}_q} \Pr[\mathbf{l}_q|k] \cdot \log_2 \Pr[k|\mathbf{l}_q] = \mathop{\mathbf{E}}_{k}\ \mathbf{H}^q_{k,k},$$

where $\mathbf{E}$ denotes the mathematical expectation and $\Pr[k|\mathbf{l}_q]$ is derived from the Bayes law. We note that this definition is equivalent to the classical one since:

$$\begin{aligned}\mathrm{H}[K|\mathbf{L}_q] &= -\sum_{\mathbf{l}_q} \Pr[\mathbf{l}_q] \sum_{k} \Pr[k|\mathbf{l}_q] \cdot \log_2 \Pr[k|\mathbf{l}_q] \\ &= -\sum_{k} \Pr[k] \sum_{\mathbf{l}_q} \Pr[\mathbf{l}_q|k] \cdot \log_2 \Pr[k|\mathbf{l}_q]\end{aligned}$$

[3] Each trace contains only one leakage sample, *i.e.* we only consider univariate attacks.

Then, we define an entropy reduction matrix: $\widetilde{\mathbf{H}}^q_{k,k^*} = \mathrm{H}[K] - \mathbf{H}^q_{k,k^*}$, where $\mathrm{H}[K]$ is the entropy of the key byte K before any side-channel attack has been performed: $\mathrm{H}[K] = -\mathbf{E}_k \log_2 \Pr[k]$. It directly yields the mutual information:

$$\mathrm{I}(K;\mathbf{L}_q) = \mathrm{H}[K] - \mathrm{H}[K|\mathbf{L}_q] = \mathop{\mathbf{E}}_{k} \; \widetilde{\mathbf{H}}^q_{k,k} \tag{2}$$

Security metric. We consider a side-channel key recovery adversary of which the aim is to guess a key byte k with non negligible probability. For this purpose and for each candidate k^*, he compares the actual observation of a leaking device $\mathbf{l}_q$ with some key dependent model for these leakages $\mathsf{M}(k^*,.)$. The construction of these models (otherwise said templates or stochastic models) will be detailed in the next section. Let $\mathsf{T}(\mathbf{l}_q, \mathsf{M}(k^*,.))$ be the statistical test used in the comparison. We assume that the highest value of the statistic corresponds to the most likely key candidate. For each observation $\mathbf{l}_q$, we store the result of the statistical test T in a vector $\mathbf{g}_q = \mathsf{T}(\mathbf{l}_q, \mathsf{M}(k^*,.))$ containing the key candidates sorted according to their likelihood: $\mathbf{g}_q := [g_1, g_2, \ldots, g_{|\mathcal{K}|}]$ (*e.g.* in our present context $|\mathcal{K}|$=256). Then, for any side-channel attack exploiting a leakage vector $\mathbf{l}_q$ and giving rise to a result $\mathbf{g}_q$, we define the success function of order o against a key byte k as: $\mathsf{S}^o_k(\mathbf{g}_q)$=1 if $k \in [g_1, \ldots, g_o]$, else $\mathsf{S}^o_k(\mathbf{g}_q)$=0. It leads to the o^{th}-order success rate:

$$\mathbf{Succ}^o_K = \mathop{\mathbf{E}}_{k} \; \mathop{\mathbf{E}}_{\mathbf{l}_q} \; \mathsf{S}^o_k(\mathbf{g}_q) \tag{3}$$

Intuitively, a success rate of order 1 (*resp.* 2) relates to the probability that the correct key byte is sorted first (*resp.* among the two first ones) by the adversary.

4 Description of the attacks

4.1 Classical template attacks

Templates construction. Suppose that an adversary is provided with N_x leakage traces corresponding to the computation of a secret value v. As will be discussed in Section 4.3, this value can but does not have to be the secret key k. In theory, one can build templates for any intermediate value computed by a leaking cryptographic device. In the template attacks of [1], a multivariate Gaussian noise is considered, which means that the vectors $\{\mathbf{l}^{v,i}_q\}^{N_x}_{i=1}$ are assumed to be drawn from the multivariate distribution:

$$\mathcal{N}(\mathbf{l}^{v,i}_q|\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v) = \frac{1}{(2\pi)^{\frac{N}{2}}|\boldsymbol{\Sigma}_v|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{l}^{v,i}_q - \boldsymbol{\mu}_v)^\top \boldsymbol{\Sigma}^{-1}_v (\mathbf{l}^{v,i}_q - \boldsymbol{\mu}_v)\right\},$$

where the mean $\boldsymbol{\mu}_v$ and the covariance matrix $\boldsymbol{\Sigma}_v$ specify completely the noise distribution associated to each secret v. Constructing the templates consists then in estimating the sets of parameters $\{\boldsymbol{\mu}_v\}^{|\mathcal{V}|}_{v=1}$ and $\{\boldsymbol{\Sigma}_v\}^{|\mathcal{V}|}_{v=1}$. A standard approach is to use the empirical mean and covariance matrix associated to the observations $\{\mathbf{l}^{v,i}_q\}^{N_x}_{i=1}$: $\hat{\boldsymbol{\mu}}_v = \frac{1}{N_x}\sum^{N_x}_{i=1} \mathbf{l}^{v,i}_q$, $\widehat{\boldsymbol{\Sigma}}_v = \frac{1}{N_x}\sum^{N_x}_{i=1}(\mathbf{l}^{v,i}_q - \hat{\boldsymbol{\mu}}_v)(\mathbf{l}^{v,i}_q - \hat{\boldsymbol{\mu}}_v)^\top$.

Attack. Assume now that there are $|\mathcal{V}|$ possible secret values. In order to determine by which secret signal a new vector $\mathbf{l}_{\text{new}}$ was generated, we apply Bayes' rule. This leads to the following classification rule:

$$\tilde{v} = \underset{v^*}{\operatorname{argmax}}\ \hat{\Pr}[v^*|\mathbf{l}_{\text{new}}] = \underset{v^*}{\operatorname{argmax}}\ \hat{\Pr}[\mathbf{l}_{\text{new}}|v^*]\,\Pr[v^*],$$

where $\hat{\Pr}[\mathbf{l}_{\text{new}}|v^*] = \mathcal{N}(\mathbf{l}_{\text{new}}|\hat{\boldsymbol{\mu}}_{v^*}, \widehat{\boldsymbol{\Sigma}}_{v^*})$ and $\Pr[v^*]$ is the a priori probability of the value candidate v^*. The classification rule assigns $\mathbf{l}_{\text{new}}$ to the candidate v^* with the highest a posteriori probability. In general, we have $\Pr[v^*] = \frac{1}{|\mathcal{V}|}$.

Interestingly, such template attacks require N_x traces to build each of the $|\mathcal{V}|$ possible models (*i.e.* mean vectors, covariance matrices). Hence, the overall number of traces for profiling q_p equals $N_x \times |\mathcal{V}|$. We note again that in our example, each execution of the S-box only gives rise to a single leakage sample. Hence we are limited to univariate attacks. But the following analysis would apply identically if each leakage trace was containing several samples.

Finally, in the (frequent) case where the values v for which the templates are built are not equal to the target key k, the adversary additionally combines the leakages corresponding to different key-dependent values in order to perform a key recovery, *i.e.* he computes $\tilde{k} = \underset{k^*}{\operatorname{argmax}}\ \prod_{i=1}^{q} \hat{\Pr}[l_{\text{new},i}|x_i, k^*]$.

4.2 Stochastic models

The stochastic models introduced in [5] work in a slightly different fashion than classical template attacks in the sense that they attempt to take advantage of the adversary's knowledge of the target device during the profiling phase. Let $\mathbf{l}_q = [l_1, l_2, \ldots, l_q]$ be the leakage vector defined in the previous sections, l_i be a leakage trace and $l_i(t)$ a leakage sample in this trace. In theory, any of those samples is the output of a leakage function L_t such that, *e.g.* in our block cipher context, $l_i(t) = \mathsf{L}_t(x_i, k)$. Stochastic models assume that this leakage function can be written as the sum of a deterministic part and a random part, namely: $\mathsf{L}_t(x_i, k) = \delta_t(x_i, k) + \rho_t$. From this basic assumption results the fact that the profiling phase will now be divided in two parts in order to approximate the leakage function deterministic part and random part separately.

Approximation of the leakage function deterministic part. In this first phase, it is assumed that the deterministic part of the leakage function can be approached as a linear combination $\hat{\delta}_t(x_i, k) = \sum_{j=0}^{u-1} \beta_{j,t} \cdot g_{j,t}(x_i, k)$, for some well chosen base functions $g_{j,t}$ of the plaintext and the key[4]. Hence, the goal of this first phase is to find the closest approximation of this form. Finding a good base $[g_{0,t}, g_{1,t}, \ldots, g_{u-1,t}]$ is typically where engineering intuition can be exploited since one has to select the functions of which the output influences the actual leakages. The better the base vector functions are correlated with the actual

[4] ... and any other possible input, *e.g.* the masks in case of protected designs.

leakages, the better the approximation of δ_t. Quite naturally, the best situation for an adversary is to have a small basis that perfectly captures all the leakage dependencies, *i.e.* to have a fast convergence towards a good approximation.

In practice, the adversary first generates N_1 leakage traces corresponding to plaintexts x_i and keys k and builds the following matrix:

$$A = \begin{pmatrix} g_{0,t}(x_1,k) & g_{1,t}(x_1,k) & ... & g_{u-1,t}(x_1,k) \\ g_{0,t}(x_2,k) & g_{1,t}(x_2,k) & ... & g_{u-1,t}(x_2,k) \\ ... & ... & ... & ... \\ g_{0,t}(x_{N_1},k) & g_{1,t}(x_{N_1},k) & ... & g_{u-1,t}(x_{N_1},k) \end{pmatrix}$$

As mentioned in Section 2, depending on the exploitation or not of a symmetry property in the leakages, it can be necessary or not to actually change the key during the profiling (note that is generally true for template attacks as well). Then, the adversary takes the leakage vector $\mathbf{l}_{N_1}(t) = [l_1(t), l_2(t), \ldots, l_{N_1}(t)]$ corresponding to the encryption of the same plaintexts with the same keys as in the matrix A. The approximation of δ_t can eventually be obtained by applying the least square method and simply computing the coefficients $\beta_{j,t}$ as follows:

$$\mathbf{b}_t = [\beta_{0,t}, \beta_{1,t}, \ldots, \beta_{u-1,t}] = (A^T \cdot A)^{-1} \cdot A^T \cdot \mathbf{l}_{N_1}(t)$$

Approximation of the leakage function random part. As for the previous template attacks, stochastic models assume a multivariate gaussian distribution for the random part of the leakages. In order to approximate this distribution, the adversary generates N_2 new traces and first evaluates a random vector that corresponds to the approximation error for m different time samples:

$$\mathbf{r}_m = [r_{t_1}, r_{t_2}, \ldots, r_{t_m}], \text{ with } r_{t_j} = \mathsf{L}_{t_j}(x_i,k) - \hat{\delta}_{t_j}(x_i,k)$$

From the N_2 realizations of the corresponding random variable R_m, he then computes the $m \times m$ empirical covariance matrix C such that $c_{ij} = Cov(r_{t_i}, r_{t_j})$.

Attack. In this third phase, the adversary obtains N_3 new traces $l_{\text{new},i}$=$\mathsf{L}(x_i,k)$. For each of those traces, he first computes a noise vector: $\mathbf{z}_i = [l_{\text{new},i}(t_1) - \hat{\delta}_{t_1}(x_i,k), l_{\text{new},i}(t_2) - \hat{\delta}_{t_2}(x_i,k), \ldots, l_{\text{new},i}(t_m) - \hat{\delta}_{t_m}(x_i,k)]$. From this vector, he can compute the following probabilities:

$$\hat{\Pr}[\mathbf{z}_i|x_i,k^*] = \frac{1}{\sqrt{(2\pi)^m|C|}} \exp\left\{-\frac{1}{2}\mathbf{z}_i^T C^{-1} \mathbf{z}_i\right\}$$

Finally, he combines these probabilities and applies the maximum likelihood rule:

$$\tilde{k} = \underset{k^*}{\operatorname{argmax}} \prod_{i=1}^{N_3} \hat{\Pr}[\mathbf{z}_i|x_i,k^*]$$

Hence, the total number of traces for profiling a stochastic model equals $q_p = N_1 + N_2$ and the number of traces in the online phase equals $q = N_3$.

4.3 Selection of templates and base vectors

A consequence of the previous descriptions is that both template attacks and stochastic models need to do some arbitrary choices before starting to profile a device. In the context of template attacks, one has to define the secret values v for which templates will be built. When stochastic models are considered, one has to determine the base functions. Therefore, if our goal is to compare those techniques on a fair basis, it is important to perform this arbitrary choice with assumptions as close as possible. For the template attacks, because we assume an EIS property, we decided to build templates for each of the $|\mathcal{V}| = 256$ possible values of $x_i \oplus k$. For the stochastic models, we assumed that the leakages were dependent of the 8 bits of the S-box output $y_i = \mathsf{S}(x_i \oplus k)$. Hence the base functions used in our experiments were $[1, y_i(1), y_i(2), \ldots, y_i(8)]$, where $y_i(j)$ denotes the j^{th} bit of y_i (here interpreted as a real number). We note that for both attacks, we could similarly assume that the leakages only depend on the Hamming weight of the S-box output. It would have resulted in the construction of only 9 templates corresponding to those Hamming weights and the use of a 2-dimensional basis $[1, H_W(y_i)]$ for the stochastic models.

5 Experiments

5.1 Empirical computation of the metrics

In this section, we present the results of different simulated profiled attacks. For this purpose, we empirically evaluated our different metrics as follows.

1. We generated large amounts of profiling traces.
2. We split these traces in different sets of q_p traces (with $N_1 = N_2 = q_p/2$)[5].
3. For various q_p values, we constructed templates and stochastic models.
4. Eventually and for various number of traces q, we evaluated the attacks, *i.e.*
   - We evaluated the probabilities $\hat{\Pr}[k^* | x_i, l_{\text{new},i}]$,
   - From those probabilities, we estimated the first-order success rate in function of q and the conditional entropy $\hat{\mathrm{H}}[K|\mathbf{L}_1]$.

In practice, the traces were generated from uniformly distributed plaintexts. We mention that since all our experiments are simulated, we were not limited in the amount of traces generated nor by statistical sampling problems in the estimation of the metrics. Note also that, following the analysis in [7], the success rate was estimated in function of the number of queries in the online phase of the attacks. By contrast, the conditional entropy was only estimated for $q = 1$.

[5] Usually, N_2 should increase as the number m of considered time instants $t_1, \ldots, t_m$ increases, while m is irrelevant for the choice of N_1. Also, if the implementation has no symmetries, N_2 is generally of subordinate relevance compared to N_1.

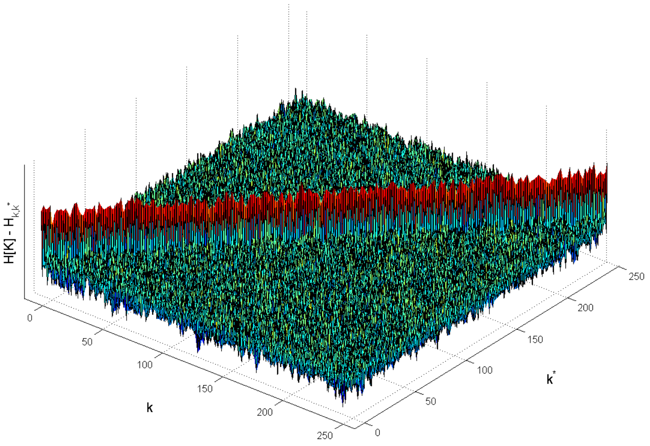

Fig. 1: Entropy reduction matrix of a sound leakage model.

5.2 Sanity check: the conditional entropy matrix

A first interesting step in the evaluation of a leakage model (*i.e.* templates-based or stochastic) is to check if it is at least good enough to perform a successful key recovery. The conditional entropy matrix is a particularly useful tool with this respect. As demonstrated in [7], Theorem 1, a matrix $\hat{\mathbf{H}}^{1}_{k,k^*}$ such that its diagonal values are minimum for all keys indicates a sound leakage model (*i.e.* a leakage model that allows asymptotically successful key recoveries). Hence, any time we constructed a new leakage model, we checked its soundness. For example, Figure 1 illustrates the entropy reduction matrix of a sound leakage model obtained from a template attack in which every template was profiled with 16 traces. We can clearly see the significantly higher information leakages of the diagonal values. It is interesting to observe that while a sound leakage model guarantees a successful key recovery, it is not a necessary condition. One could easily imagine a leakage model such that only certain templates have been properly profiled and nevertheless leads to successful attacks.

5.3 Evaluation of the attacks

Next to the sanity check of the conditional entropy matrix, Figures 2 and 3 respectively represent the estimation of our metrics for the two considered attacks. Interestingly, the success rate plot is 2-dimensional since it depends on both q_p and q. By contrast, the conditional entropy plot is only computed for $q = 1$ and hence only depends on q_p. Quite naturally, the success rate tends to one when the number of traces in the profiling and online phases increases. It is worth noting that the conditional entropy value is sometimes higher than 8 which clearly indicates an insufficient profiling. From a practical point of view, the figure directly suggests the increased effectiveness of the profiling phase when using stochastic models compared to template attacks. This is because only one function in a

9-dimensional subspace has to be approximated compared to the building of 256 templates. From a more theoretical point of view, we can see that an increase in the amount of traces for profiling improves the effectiveness of the attacks (or informativeness of the models) up to a certain bound. It is consequently interesting to use the information theoretic metric to determine this bound. In our example, we can observe that template attacks and stochastic models have their conditional entropy that seem to converge towards the same value. It indicates that the base functions used to approximate the leakages properly capture their dependencies (which is expected since we know that the leakages actually correspond to the noisy Hamming weights of the AES S-box output).

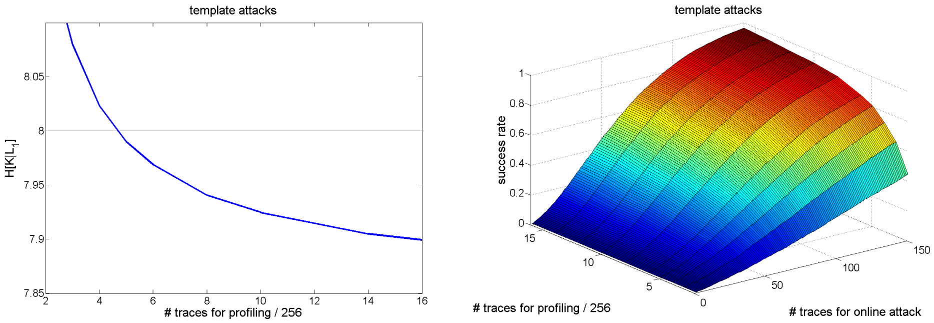

Fig. 2: Conditional entropy and success rate of the template attacks.

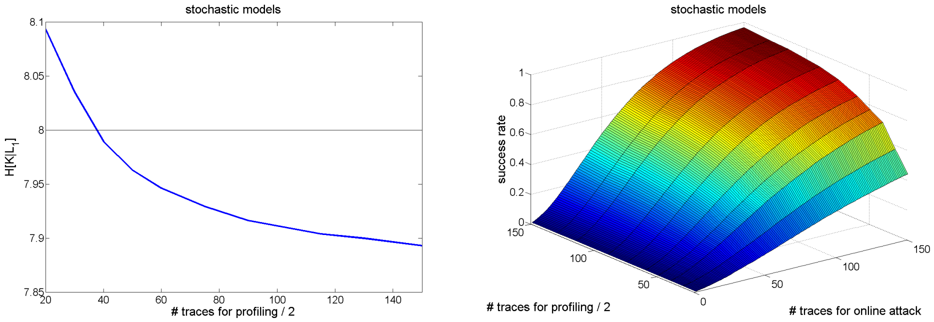

Fig. 3: Conditional entropy and success rate of the stochastic models.

5.4 Comparison of the attacks

Given the previous results, a very natural question is to wonder if we can properly quantify the effectiveness of the profiling and online phases of the investigated attacks. As a matter of fact, this question can be divided in three parts: (1) "which profiling is the fastest to build a sound model?", (2) "which profiling gives rise to the smallest conditional entropy?" and (3) "which profiling gives rise to the most efficient online attacks?". In order to answer these questions, it is convenient to plot the conditional entropy values in a logarithmic scale as in the left part of Figure 4. From this picture, it clearly appears that the profiling of stochastic models is one order of magnitude faster than the one of classical template attacks in our example, which answers the first question. We then see (again) that both methods seem to converge towards the same conditional entropy value which answers the second question. Eventually and following [7], Theorem 2, this also implies that stochastic models and template attacks should be as efficient in the online attacks if a sufficient profiling is used. This is because a more informative model generally gives rise to a more efficient online attack. As an illustration, we plotted the success rates corresponding to three different profiling phases in the right part of the figure and they confirm this intuition. Hence, the information theoretic and security metrics appear as good methods for the comparison of the profiling and online attack efficiencies, respectively. In practice, since the main goal of a profiling step is to build a precise leakage model, the most important parameter to compare this step is usually the smallest value of the conditional entropy that can be reached. But when the limit of this conditional entropy for increasing q_p values is identical for different methods or in contexts where the number of profiling traces is limited, the rapidity of converging towards a sound leakage model becomes important as well.

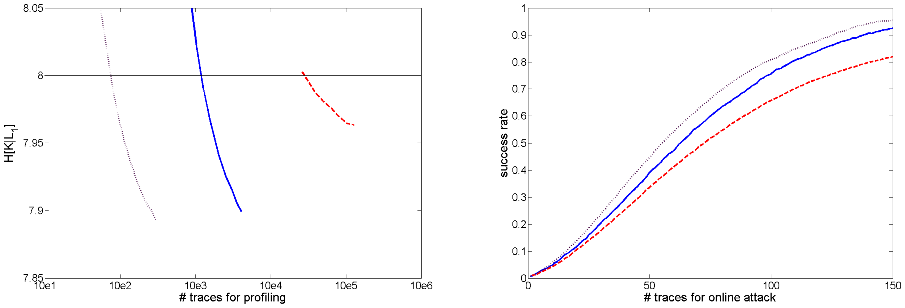

Fig. 4: Left figure: comparison of the conditional entropies - plain: template attacks, dotted: stochastic models, dashed: histograms. Right figure: comparison of the success rates - plain: template attack with $\mathrm{H}[S|\mathbf{L}_1] \simeq 7.94$, dotted: stochastic model with $\mathrm{H}[S|\mathbf{L}_1] \simeq 7.92$, dashed: stochastic model with $\mathrm{H}[S|\mathbf{L}_1] \simeq 7.98$.

Summarizing, the speed of convergence of a profiling method is measured by the X axis in the left part of Figure 4; the informativeness of the profiled models is measured by the Y axis of Figure 4; and this informativeness is generally related to the success rate of the corresponding online attacks.

We mention that for illustration, we also evaluated a naive profiling in which the Gaussian templates were replaced by histograms. As observed in the left part of Figure 4, such histograms are slower to build less informative models. In theory, one could of course imagine many other types and contexts of profiling (*e.g.* profiling that produces sound but not very informative models very fast or profiling that produces very informative models very slowly).

6 Limitations

The previous sections were dedicated to the description of an exemplary context in which the proposed methodology to compare profiled side-channel attacks was meaningful. Before concluding the paper, this section aims to briefly discuss the extent to which the previous conclusions are generally true.

A first restriction that has to be mentioned relates to the evaluation framework itself. As demonstrated in [7], there is no one-to-one relation between the conditional entropy and the success rate computed for a general leakage function. In numerous practical applications, the intuition that more conditional entropy implies less success rate is verified. But this does not prevent the possible existence of counterintuitive situations. It remains that the proposed metrics and relations are at least more meaningful than ad hoc evaluation criteria. But a certain level of scepticism and the verification of some relations such as in the right part of Figure 4 are always in place in the analysis of side-channel attacks.

A second restriction relates to the evaluation of the metrics in real measurement environments where statistical sampling can become an issue. As a matter of fact, reaching a high confidence level in the evaluation of the metrics when computed from small unprotected devices is generally not an issue. But, *e.g.* computing the conditional entropy for a protected hardware design can be more difficult. With this respect, it is worth remembering that comparing implementations according to their information leakages is only meaningful in the context of sound leakage models. Hence, the more challenging the target device, the more interesting the entropy matrix sanity check of Section 5.2.

Thirdly, it is important to acknowledge that the comparison between two side-channel attacks such as templates attacks and stochastic models in this paper is in essence implementation-dependent. What this paper provides is a methodology that allows comparing these attacks on a sound basis, for one given implementation (or for a class of similar implementations). But changing the experimental conditions can affect the practical conclusions that are obtained from a set of experiments. For example, we conclude from our investigated context that the profiling efficiency of stochastic models is much higher than the one of classical template attacks. In fact, this conclusion mainly holds because the base functions chosen to build our stochastic models perfectly capture the actual leakage dependencies. But in case the base vectors are not perfectly chosen, the generic nature of template attacks may allow them to better incorporate the physical specificities of the measurements. Yet, we point out that the subspace can always be selected so large that it catches all relevant peculiarities, possibly at cost of profiling efficiency. Hence, template attacks can be viewed as the limiting case of the stochastic approach, when the subspace equals the full vector space. In other words, stochastic models generally trade a bit of the generality of template attacks for a more efficient (*i.e.* faster) profiling.

Eventually, let us mention that there are situations where templates are more appropriate than stochastic models. An interesting example is the following. Say the S-box in a block cipher is unknown and a device only leaks the Hamming weights of this S-box output. Then, templates can still be built for any value of $x_i \oplus k$ and result in a sound leakage model. By contrast, the previous (standard) selection of basis vectors that depend on $\mathsf{S}(x_i \oplus k)$ is not possible anymore. And a basis made of the 8 bits of $x_i \oplus k$ will not lead to a good approximation of the leakage function, because it does not not capture the S-box non-linearity.

7 Conclusions

This paper presents an application of the methodology introduced in [7] to the analysis of template attacks and stochastic models. We investigated an exemplary context of simulated leakages in order to confirm the soundness of some metrics to compare profiled side-channel attacks. Extending this analysis and evaluation towards more complex scenarios is a good scope for further research.

In particular, the evaluation of multivariate attacks against masked implementations or non-CMOS devices would be interesting. Since in general, the problem of power-based side-channel attacks can be viewed as a probability density function estimation problem, it is expected that the intuition provided by an information theoretic analysis as in this work will generally hold. But additional empirical confirmations would strengthen this expectation. For example, it is known that the conditional entropy can be used to evaluate masked implementations [8] and that stochastic models are also applicable in this context [3, 6]. A practical question is to determine how much the masking exactly affects the profiling efficiency of profiled attacks (with known or unknown masks).

Acknowledgements. The authors would like to thank the reviewers of ACNS 2009 for their meaningful comments about this work.

References

1. S. Chari, J. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, Lecture Notes in Computer Science, vol 2523, pp 13-28, CA, USA, August 2002.
2. B. Gierlichs, K. Lemke, C. Paar, *Templates vs. Stochastic Methods*, in the proceedings of CHES 2006, Lecture Notes in Computer Science, vol 4249, pp 15-29, Yokohama, Japan, October 2006.
3. K. Lemke, C. Paar, *Analyzing Side-Channel Leakage of Masked Implementations with Stochastic Methods*, in the proceedings of ESORICS 2007, Lecture Notes in Computer Science, vol 4734, pp 454-468, Dresden, Germany, September 2007.
4. S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks*, Springer, 2007.
5. W. Schindler, K. Lemke, C. Paar, *A Stochastic Model for Differential Side-Channel Cryptanalysis*, CHES 2005, Lecture Notes in Computer Science, vol 3659, pp 30-46, Edinburgh, Scotland, September 2005.
6. W. Schindler, *Advanced Stochastic Methods in Side-Channel Analysis on Block Ciphers in the Presence of Masking*, J. of Math. Cryptology, vol 2, pp 291-310, 2008.
7. F.-X. Standaert, T.G. Malkin, M. Yung, *A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks*, to appear in the proceedings of Eurocrypt 2009. Extended version available from: Cryptology ePrint Archive, Report 2006/139.
8. F.-X. Standaert, E. Peeters, C. Archambeau, J.-J. Quisquater, *Towards Security Limits in Side-Channel Attacks*, in the proceedings of CHES 2006, Lecture Notes in Computer Science, vol 4249, pp. 30–45, Yokohama, Japan, October 2006. Latest version available from: http://eprint.iacr.org/2007/222.
9. F.-X. Standaert, C. Archambeau, *Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages*, CHES 2008, Lecture Notes in Computer Science, vol 5154, Washington DC, USA, August 2008.